Performance Analysis of Machine Learning Techniques in Network Intrusion Detection

Md. Biplob Hosen[1], Ashfaq Ali Shafin[2] and Mohammad Abu Yousuf[1]

[1]Institute of Information Technology, Jahangirnagar University, Savar, Dhaka-1342, Bangladesh
[2]Florida International University
biplob.hosen@juniv.edu

Abstract. A lot of sensitive data is being transmitted over the internet nowadays, which leads to increased risks of network attacks. To identify suspicious and malicious activities to secure internal networks, intrusion detection systems aim to recognize unusual access or attacks to the network. Machine learning technology can play a vital role in a scheme to detect intrusion. It is a technology that is based on classification and prediction, to deal with security threats. In this work, we focus on significant feature selection and classification using four machine learning algorithms. Adaptive Boost (AdaBoost), Gradient Boosting, Random Forest, and Decision Tree classification techniques have been tested on the dataset of network intrusion detection which is collected from Kaggle. In our analysis, Gradient Boosting outperforms considering the F1-score. Therefore, this machine learning technique can be utilized to implement an intelligent intrusion detection system.

Keywords: Intrusion Detection, F1-score, AdaBoost, Gradient Boosting, Random Forest, Decision Tree.

1 Introduction

At present, the popularity of using the internet has significantly increased. People are now dependent on the internet in mostly every aspect of their life. In our daily life, numerous amount of data is being transmitted over the internet for different purposes such as education, healthcare, entertainment, etc. Among them, some of the data are highly sensitive and critical. So, protection of the transmitted information is now a major concern. Attackers always target to destroy the confidentiality, integrity, and availability of a system which are indicated as CIA property. Any attempt to break this CIA property of a system is called intrusion [1]. To block internet-based attacks, an intrusion detection system (IDS) performs a significant contribution. Nowadays, applications based on wireless sensor networks (WSN) are gaining a colossal popularity index. Due to resource limitations, it is not always conceivable to implement effective security management techniques in WSN. In these types of situations, IDS is the ideal solution to resist external attacks. It ensures a wall of defense to prevent any abnormal behavior in the network. IDS can work smoothly in cases where a conventional firewall fails to perform the task. IDS assumes that the practice of intruders is different from the behavior of a regular user. Anomaly-based and misuse-based are two classes of IDSs [2]. IDS peruses the pattern from training set data to identify only the well-known data in misuse-based detection. Anomaly-based detection monitors general activities, and if any divergence from the normal activity is noticed, it is classified as an anomaly. In this way, anomaly-based IDS can identify unknown attacks, which are not directly trained. Observation and audit of the occurring activities are significant parts of an IDS. A host-based intrusion detection system (HIDS), network-based intrusion-detection system (NIDS), and distributed intrusion-detection system (DIDS) are three categories of IDS based on the placement of the IDS [1]. In HIDS, the host events are observed to detect intrusion inside of the host. To detect intrusion inside of a network, NIDS observes and monitors the network activities. In DIDS, many HIDS and NIDS work together to inspect activities over an extensive network. Anomaly-based IDS, signature-based IDS, and hybrid IDS are some strategies for network intrusion detection. Another popular approach is the data-driven method which mostly focuses on minority attack classes compared to normal traffic [3].

In our study, we perform the classification of anomalies on the publicly available Kaggle dataset of network intrusion detection. Firstly, we use all 41 features to train and classify using Adaptive Boost (AdaBoost), Gradient Boosting, Random Forest, and Decision Tree classifiers and identify the most significant 15 features for each classifier. For each technique, performance is studied, based on 5-fold cross-validation. Then we use these significant features to reduce training and testing time with almost the same performance. In this paper, firstly, we have discussed some relative works in Section II. The working methodology is presented in Section III. The result is discussed in Section IV. Finally, we conclude in Section V.

2 Literature Review

Due to the rapid change like malicious threats, advanced security features are required in networks, and Khan et al. proposed a hybrid ID framework to classify the malicious threats [4]. The proposed model is a combination of CNN and RNN where CNN helps to capture local features and RNN provides a changing attack detection method. Andresini et al. considered intrusion detection as a binary classification problem and proposed a DML methodology, which is a combined method with Autoencoders and Triplet networks [5]. They claimed that their model outperforms the imbalance of network traffic data. Kan et al. emphasized the detection of intrusion on IoT networks and proposed an adaptive particle swarm optimization-CNN-based intrusion detection method [6]. According to Zhang et al. [7], data modalities and mutual support among various data attributes are to be considered to build a credible IDS. Their proposed model is based on multi-dimensional feature fusion and stacking ensemble mechanisms. They claimed that their model is an effective one to identify any abnormal behavior. Bagui et al. conducted a study on six commonly used datasets of network intrusion detection and claimed that traditional artificial neural network models (ANN) cannot effectively identify minority attacks for imbalanced datasets [8]. According to them, "Resampling" can be a solution, as they found that oversampling and undersampling increase recall and accuracy to detect minority attacks. The lack of a dataset with standard features is another major problem to generalize different network scenarios according to Sarhan et al. [9]. They generated five new datasets merging others using NetFlow-based standard feature sets. They tested their dataset using an Extra tree classifier both for binary and multi-classification and claimed to achieve higher detection performance compared to proprietary feature sets. Wang et al.identified the limitations of traditional training models like back propagation and proposed an improved deep belief network model based on a kernel-based extreme learning machine where an enhanced grey wolf optimizer is designed to optimize kernel parameters [10]. According to their study, their model outperforms in terms of confusion matrices on the commonly used datasets. To find the cause of the problems associated with different machine learning algorithms to identify network intrusion, Mishra et al. [11] presented a detailed investigation. They mapped each attack with corresponding features. Mueller et al. [12] proposed a real-time framework for intrusion detection. Their model is focused on timing analysis using machine learning algorithms. They used a false negative (FN) rate to define undetected anomalies and a false positive (FP) rate to indicate normal states which are flagged as abnormal. For the analysis of enormous traffic, an efficient classification model is required according to Ahmad et al. [13]. They applied SVM, Random Forest, and Extreme Learning Machine (ELM) to NSL-knowledge discovery and data mining datasets. They mentioned that ELM outperforms in their analysis. Shone et al. [14] presented a deep learning technique to detect intrusion. They implemented their model in GPU-enabled Tensor-Flow and tested the performance using benchmark KDD cup'99 and NSL-KDD datasets. Sharafaldin et al. [15] have described that most of the existing dataset of intrusion detection is not suitable as the traffic diversity and volumes are not well enough there; some of them do not also cover the diversity of attacks. They produced their dataset which contains seven frequent attacks. They used the Random Forest algorithm to extract the best features. They evaluated the performance using seven machine learning algorithms. They concluded by comparing performance with some available datasets. The effectiveness of two open-source IDSs named Snort and Suricata were investigated and compared by Shah et al. [16]. Snort showed better accuracy in intrusion detection but had a high false-positive rate. An optimized SVM with a firefly algorithm helped to achieve the best result. Aljawarneh et al. [17] developed a hybrid model that has two parts. Data are filtered using the Vote algorithm with information gained in the first part to select relevant features. The second part contains a hybrid model including J48, Meta paging, RandomTree, REPTree, AdaBoostM1, DecisionStaump, and Naïve Bayes. Mohammadi et al. [18] have proposed an end-to-end deep adversarial network architecture. Their architecture consists of two deep networks. They evaluated their model on the NSL-KDD dataset to analyze performance. A multilevel intrusion detection model named multilevel semi-supervised machine learning (MSML) has been proposed by Yao et al [19]. They claimed that to recognize unknown patterns their model is superior to another existing model. Sinclair et al. [20] employed genetic algorithms and a decision tree to classify network connections. They generated rules using these methods of classification. They concluded that methods like 'neural networks' can improve their model to detect sophisticated attacks. Hajimirzaei et al. [21] proposed an IDS that is based on the combination of multilayer perceptron (MLP), fuzzy clustering, and artificial bee colony (ABC). They used MLP to identify abnormal network traffic packets. The fuzzy clustering method provided different training subsets. ABC algorithm is used to update the values for linkage weights and biases which are necessary to train the MLP. They evaluated the performance of their proposed model using the cloudSim simulator and found that their model can correctly classify instances with 98.42% accuracy. Kaja et al. [22] proposed a two-stage architecture of an IDS system to identify malicious attacks. In their architecture, initially, they utilized a k-means algorithm to detect attacks. In the next stage, they used different supervised learning classifiers to classify attacks. They claimed that they acquired 99.97% accuracy with their model. Malhotra et al. [23] discussed the importance of feature selection as it requires a lot of time to build a model with all features. They evaluated the performance of ten classification algorithms on the NSL-KDD dataset and found that Random Forest, Bagging, PART, and J48 are the best classifiers in terms of performance. But they consumed a lot of time to build a model. To minimize build time, they used two approaches, namely, the 'wrapper method' and 'filter method' for feature selection and feature reduction. With the selected features only, they again ran the top four classification algorithms on the dataset and found significant performance with a smaller build time. Ling et al. [24] proposed a model of IDS, where they use the Random Forest feature selection algorithm on the dataset collected from KDD Cup99. Using the optimal features, they trained their model which is a multi-classifier ensemble model based on deep learning. The assembled model included SVM, Bayesian, Decision tree, and K-NN classifiers. They found that their model performs better than the combined model of Random Forest and majoring voting ensemble method. Khorram et al. [25] havediscussed the importance of false alarms mitigation in network intrusion detection. They used partial swarm optimization (PSO), ant colony optimization (ACO), and Artificial Bee Colony (ABC) algorithms to select the most significant features. They evaluated the performance of these feature selection methods using K-NN and SVM classifiers. Performance evaluation on the NSL-KDD dataset shows that the K-NN classifier with the ABC feature selection method provides the highest accuracy compared to five other models. Aslahi-Shahri et al. [26] introduced a GA-SVM model of anomaly detection. The genetic algorithm (GA) is used for feature selection to select ten significant features out of forty-five features. Using the selected features and SVM classifiers, they claimed to achieve a 97.3% true-positive rate. Jabez et al. [27] proposed a model to detect anomaly which is an aggregation of correlation-based feature selection (CFS), extreme learning, and Multilayer perceptron. Anwer et al. [28] presented an efficient feature selection framework. They used filter and wrapper methodologies of feature selection. They applied these techniques to the UNSW-NB15 dataset and found 88% accuracy with the J48 classifier using the top eighteen features.

3 Methodology

3.1 Dataset Description

The dataset is collected from the Kaggle network intrusion detection module, which contains a wide variety of intrusions. There are 41 features, 3 of them are qualitative and 38 are quantitative. The class variable has two types such as: normal and anomalous. An overview of the dataset is presented in Table 1 which includes an illustration of the dataset.

Table 1. Dataset Description.

Class	Total Data	Training Data	Testing Data
Normal	13449	9389	4060
Anomaly	11743	8245	3498

3.2 Data Preprocessing

In this phase, some preprocessing on the collected dataset is performed to make data suitable for analysis. Using the mean/median of other values, the missing values are filled up. It is also ensured that no duplicate values are there in the dataset.

3.3 Apply Machine Learning Algorithms

We have applied four popular machine learning techniques of classification to identify the 15 most important features for each classifier and to detect network intrusion. An overview of four techniques is given below:

A) AdaBoost: AdaBoost is a technique formulated by Yoav Freund and Robert Schapire. This model combines some weak classifiers, corrects misclassifications made by weak classifiers, and constructs a strong classifier. Network intrusion detection using AdaBoost consists of four building blocks such as: extracting features, labeling data, modeling weak classifiers, and developing the strong classifier [29]. Firstly, equal weight is assigned for each of the data points in the dataset. Then, inappropriately classified points are identified. Finally, the weight of inappropriately identified data points is increased until the required results are achieved.

B) Decision Tree: A decision tree uses a tree-based approach to detect network intrusion. In this technique, features are indicated by internal nodes, rules are indicated by branches, and leaf node values are the outcomes. ID3, J48, C4.5, C5, CART, and CHAID are some well-known algorithms for decision trees [30]. In our study, the J48 method has been selected to evaluate the efficiency of the dataset.

C) Gradient Boosting: Gradient Boosting algorithms are highly customizable to particular applications. The main objective of this technique is the minimization of the loss function by combining weak learners using gradient descent. Gradient Boosting technique involves three phases i) Optimization of the loss function, ii) Prediction using weak learners, and iii) Minimization of loss function by adding weak learners. Extreme Gradient Boosting (XGBoost), Lightweight Gradient Boosting Machines (LightGBM), and CatBoost are some well-known variants of Gradient Boosting. In our study, the most popular XGBoost variant is used which is an ensemble machine learning approach.

D) Random Forest: Random Forest constructs several decision trees and combines the individual prediction of each tree using majority voting to predict the final output. Random Forest provides higher accuracy even for a large dataset with comparatively lower training time. This algorithm can ensure an acceptable level of accuracy even though a large number of data is missing.

3.4 Evaluation Metric

Precision, recall, F1-score, and accuracy are used for performance analysis. We mostly focus on F1-score for this imbalance dataset. Equation-1 helps to calculate F1-score based on precision and recall [14].

$$F1\_score = \frac{2(\mathrm{Pr}\,ecision * \mathrm{Re}\,call)}{(\mathrm{Pr}\,ecision + \mathrm{Re}\,call)} \tag{1}$$

Where precision and recall are determined by the true positive, false positive, and false negative.

Fig.1 is an illustration of the total working procedure of our study. According to the figure, initially, some pre-processing is performed on the collected dataset. Then, we have used the training dataset and four machine learning algorithms to develop models and select significant features. Finally, some analyses are performed using the test dataset, and based on the analysis, the best prediction classifier is chosen.

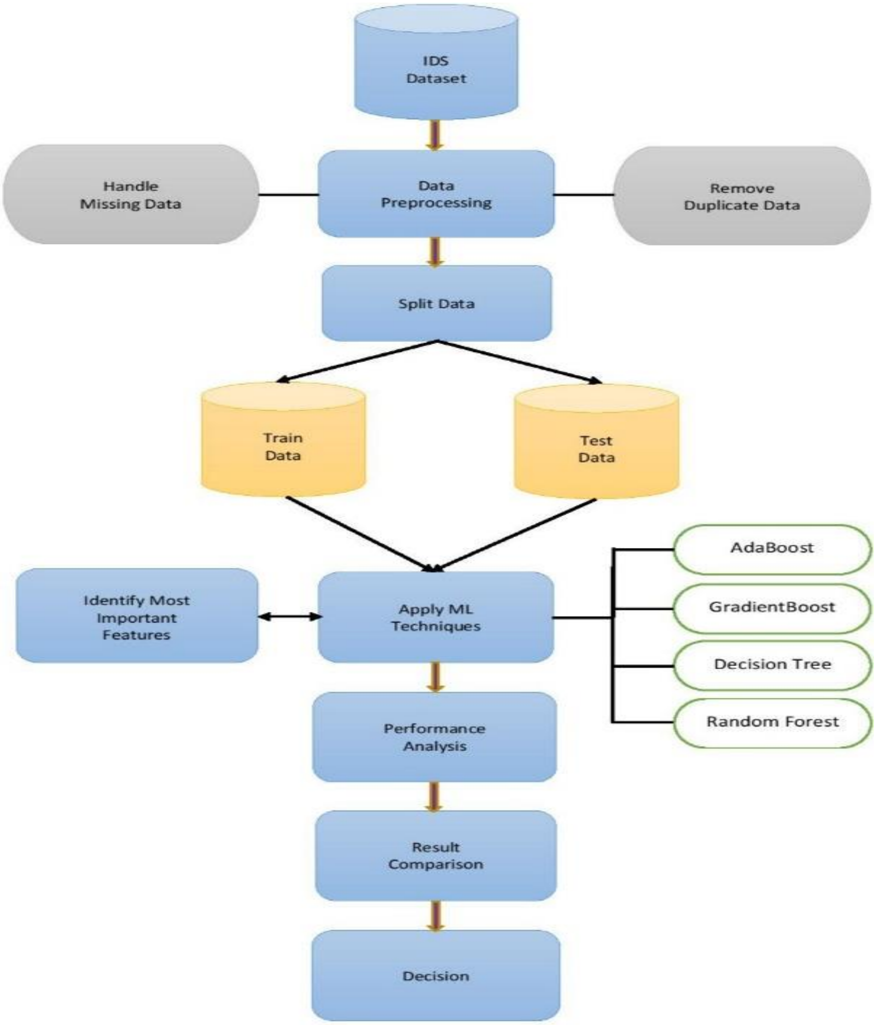

Fig. 1. Classification Workflow

4 Result Analysis & Discussion

In our study, we have used four different algorithms to decide how well our intrusion detection system can detect unknown threats. At first, we run the simulation using all 41 features for the above-mentioned four algorithms and identify the 15 most significant features for each of the classifiers. Then we again run the experiment using only the top 15 features to analyze the performance. Fig.2 illustrates significant features of four algorithms. The significance level of the features is scaled from 0 to 1. Features like src_bytes, dst_byte, flag, protocol_type, and service have a significant impact on almost every algorithm.

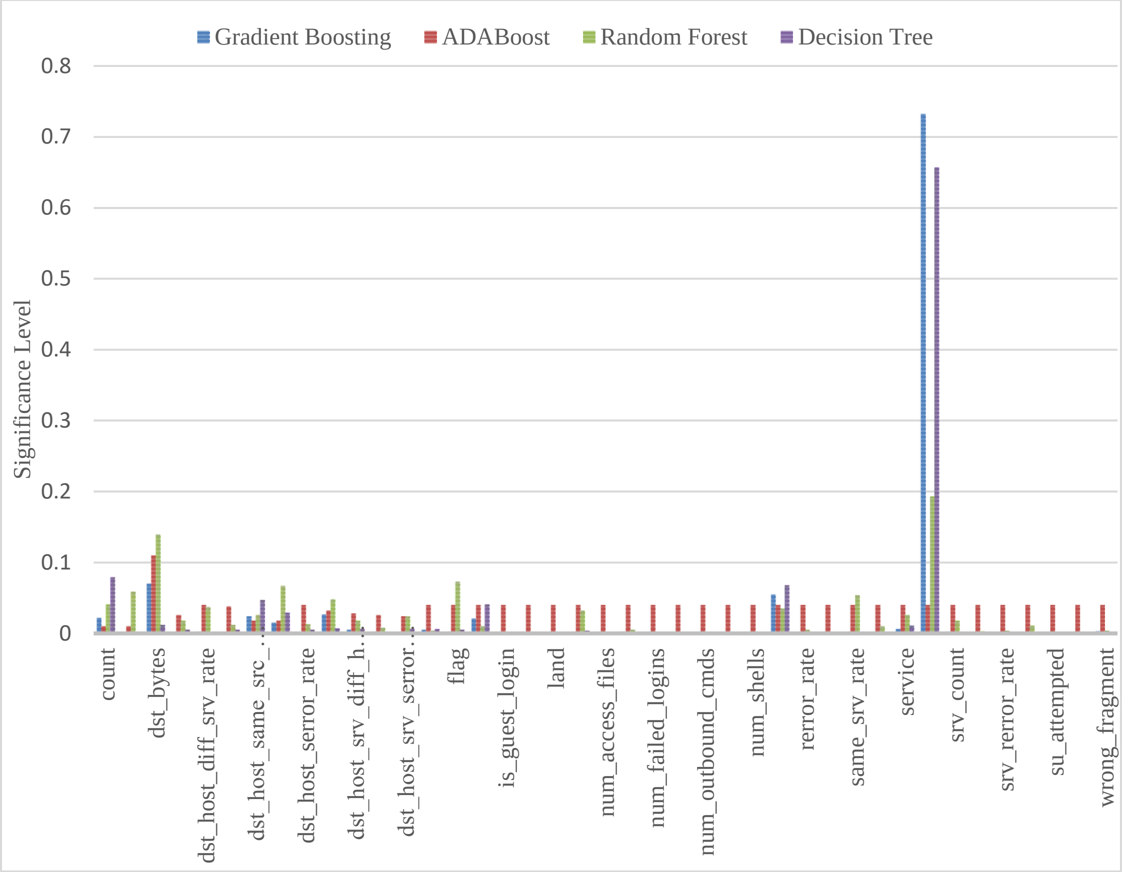

Fig. 2. Feature Importance of Our Dataset

Table 2 demonstrates matrices for four algorithms both for 41 features and 15 most important features using the training dataset and testing dataset.

Table 2. Result Analysis of Different Algorithms

		Training Data				Testing Data			
Classifier	Features Used	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
AdaBoost	41	99.47	99.48	99.46	99.47	99.58	99.49	99.60	99.54
	15	99.39	99.40	99.38	99.39	99.36	99.25	99.37	99.31
Gradient	41	99.72	99.73	99.71	99.72	99.72	99.66	99.74	99.70
Boosting	15	99.7	99.70	99.69	99.70	99.65	99.45	99.80	99.63
Random	41	99.7	99.71	99.69	99.70	99.71	99.77	99.60	99.69
Forest	15	99.65	99.66	99.64	99.65	99.58	99.57	99.51	99.54
Decision	41	99.53	99.53	99.53	99.53	99.50	99.37	99.54	99.46
Tree	15	99.49	99.45	99.44	99.45	99.56	99.40	99.66	99.53

Gradient Boosting outperforms in this dataset while accuracy is concerned for both the top 15 features model and all 41 features model. For precision, the Random Forest classifier has the best output. Again, Gradient Boosting outperforms considering both recall and F1-score. The reasons behind the effectiveness of Gradient Boosting are- learning models can correct each other's errors, and they can capture complex patterns in the dataset. Fig.3 represents the F1-score analysis of different algorithms on different feature sets. For an imbalance dataset like this one of IDS, F1-score gives the best performance comparison.

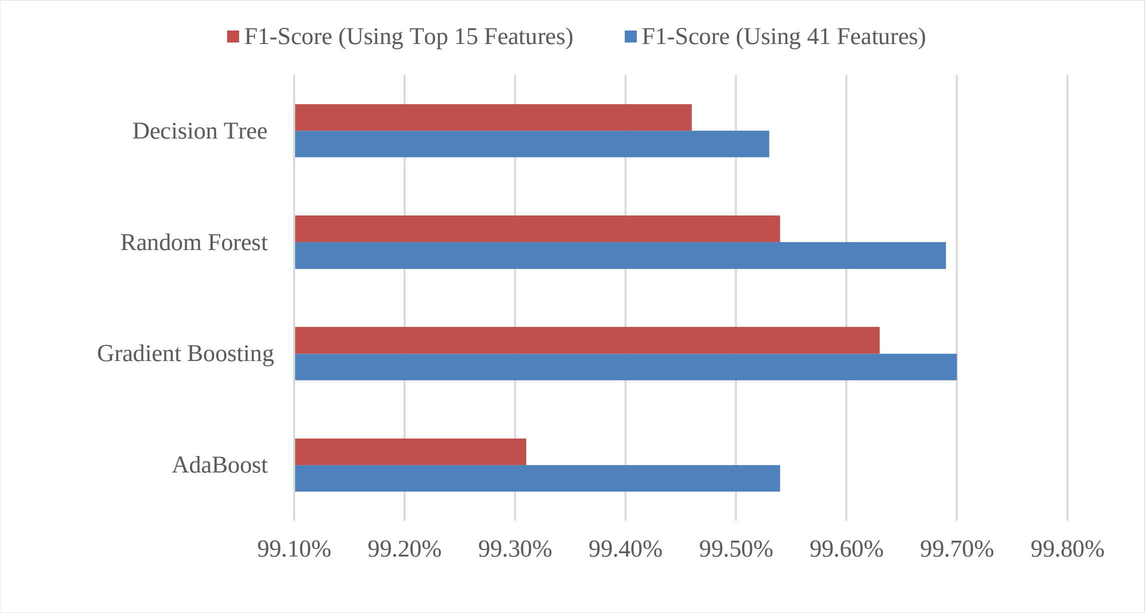

Fig. 3. F1-Score Analysis

Fig.4 demonstrates a training time (in seconds) comparison between the top 15-features model and 41-features model for four algorithms. Training time for the Gradient Boosting algorithm is very much lower for the 15-features model compared to the 41-features model.

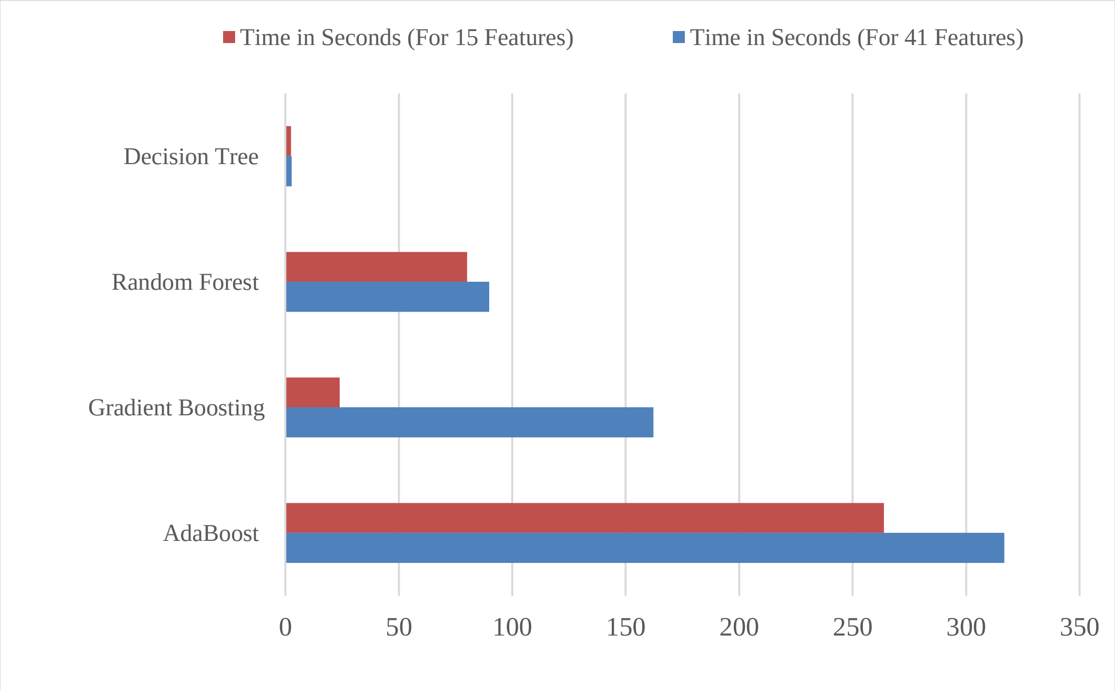

Fig. 4. Training Time Comparison

Fig.5 represents the testing time (in seconds) comparison between two models for four algorithms. Though the testing times for Decision Tree, Gradient Boosting, and Random-forest technique are almost the same for the top 15-features model and 41-features model, they are reduced significantly for the AdaBoost technique.

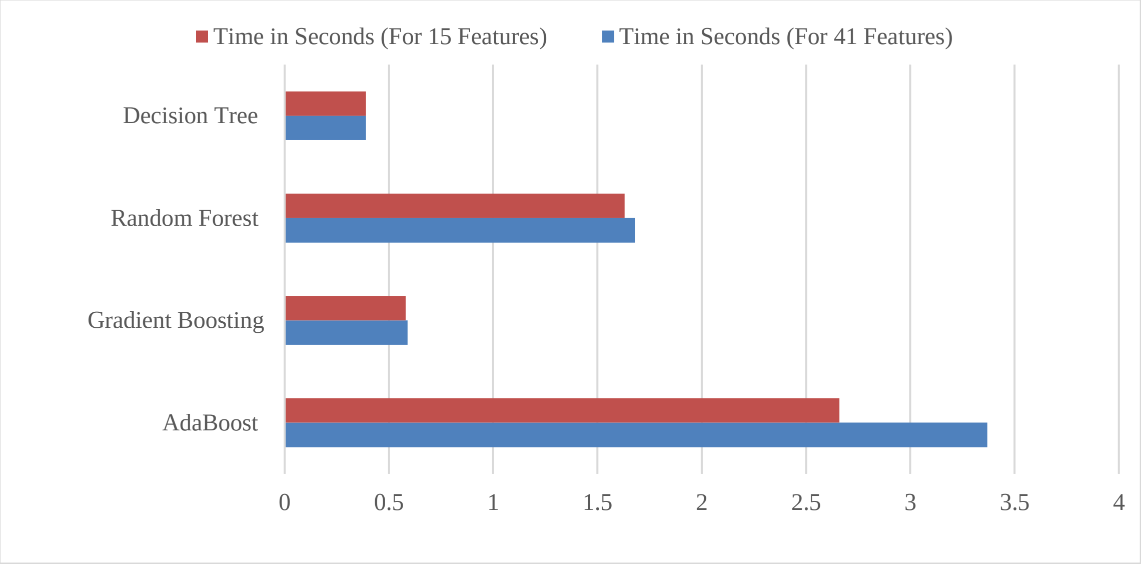

Fig. 5. Testing Time Comparison

5 Conclusion

Network intrusion detection has a vital impact on the safety of a system. We have implemented four existing machine learning algorithms, to build a model for network intrusion detection. Using all features, it usually takes more time to create a model of machine learning. Hence, it is efficient to work with significant features. Initially, we have identified the top 15 significant features using all four algorithms. We have evaluated the effectiveness using confusion matrices, and training and testing time considering both the 41-features model and top 15-features model. We conclude with the decision that Gradient Boosting outperforms in our analysis. Our imminent plan is to implement deep learning in this system. It may further be possible to design a more sophisticated classifier using a hybrid model.

References

1. Sakr, M.M., Tawfeeq, M.A. and El-Sisi, A.B., 2019. An efficiency optimization for network intrusion detection system. International Journal of Computer Network and Information Security, 11(10), p.1.
2. Belavagi, M.C. and Muniyal, B., 2016. Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Computer Science, 89, pp.117-123.
3. Chou, D. and Jiang, M., 2021. A survey on data-driven network intrusion detection. ACM Computing Surveys (CSUR), 54(9), pp.1-36.
4. Khan, M.A., 2021. HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system. Processes, 9(5), p.834.
5. Andresini, G., Appice, A. and Malerba, D., 2021. Autoencoder-based deep metric learning for network intrusion detection. Information Sciences, 569, pp.706-727.
6. Kan, X., Fan, Y., Fang, Z., Cao, L., Xiong, N.N., Yang, D. and Li, X., 2021. A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network. Information Sciences, 568, pp.147-162.
7. Zhang, H., Li, J.L., Liu, X.M. and Dong, C., 2021. Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. Future Generation Computer Systems, 122, pp.130-143.

8. Bagui, S. and Li, K., 2021. Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data, 8(1), pp.1-41.
9. Sarhan, M., Layeghy, S. and Portmann, M., 2021. Towards a standard feature set for network intrusion detection system datasets. Mobile Networks and Applications, pp.1-14.
10. Wang, Z., Zeng, Y., Liu, Y. and Li, D., 2021. Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection. IEEE Access, 9, pp.16062-16091.
11. Mishra, P., Varadharajan, V., Tupakula, U. and Pilli, E.S., 2018. A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials, 21(1), pp.686-728.
12. Xu, H. and Mueller, F., 2018, December. Machine learning enhanced real-time intrusion detection using timing information. In International Workshop on Trustworthy & Real-time Edge Computing for Cyber-Physical Systems.
13. Ahmad, I., Basheri, M., Iqbal, M.J. and Rahim, A., 2018. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE access, 6, pp.33789-33795.
14. Shone, N., Ngoc, T.N., Phai, V.D. and Shi, Q., 2018. A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), pp.41-50.
15. Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1, pp.108-116.
16. Shah, S.A.R., and Issac, B., 2018. Performance comparison of intrusion detection systems and application of machine learning to Snort system. Future Generation Computer Systems, 80, pp.157-170.
17. Aljawarneh, S., Aldwairi, M., and Yassein, M.B., 2018. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science, 25, pp.152-160.
18. Mohammadi, B. and Sabokrou, M., 2019, October. End-to-end adversarial learning for intrusion detection in computer networks. In 2019 IEEE 44th Conference on Local Computer Networks (LCN) (pp. 270-273). IEEE.
19. Yao, H., Fu, D., Zhang, P., Li, M., and Liu, Y., 2018. MSML: A Novel Multilevel Semi-Supervised Machine Learning Framework for Intrusion Detection System. IEEE Internet of Things Journal, 6(2), pp.1949-1959.
20. Sinclair, C., Pierce, L., and Matzner, S., 1999, December. An application of machine learning to network intrusion detection. In Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99) (pp. 371-377). IEEE.
21. Hajimirzaei, B. and Navimipour, N.J., 2019. Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. ICT Express, 5(1), pp.56-59.
22. Kaja, N., Shaout, A. and Ma, D., 2019. An intelligent intrusion detection system. Applied Intelligence, pp.1-13.
23. Malhotra, H. and Sharma, P., 2019. Intrusion Detection using Machine Learning and Feature Selection. International Journal of Computer Network & Information Security, 11(4).
24. Ling, J. and Wu, C., 2019, April. Feature selection and deep learning based approach for network intrusion detection. In the 3rd International Conference on Mechatronics Engineering and Information Technology.
25. Khorram, T. and Baykan, N.A., 2018. Feature selection in network intrusion detection using metaheuristic algorithms. International Journal of Advance Research, Ideas and Innovations in Technology, 4(4), pp.704-710.
26. Aslahi-Shahri, B.M., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M.J. and Ebrahimi, A., 2016. A hybrid method consisting of GA and SVM for intrusion detection system. Neural computing and applications, 27(6), pp.1669-1676.
27. Jabez, J., Gowri, S., Vigneshwari, S., Mayan, J.A. and Srinivasulu, S., 2019. Anomaly Detection by Using CFS Subset and Neural Network with WEKA Tools. In Information and Communication Technology for Intelligent Systems (pp. 675-682). Springer, Singapore.
28. Anwer, H.M., Farouk, M. and Abdel-Hamid, A., 2018, April. A framework for efficient network anomaly intrusion detection with features selection. In 2018 9th International Conference on Information and Communication Systems (ICICS) (pp. 157-162). IEEE.
29. Hu, W., Hu, W. and Maybank, S., 2008. Adaboost-based algorithm for network intrusion detection. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 38(2), pp.577-583.
30. Faruque, M.F. and Sarker, I.H., 2019, February. Performance analysis of machine learning techniques to predict diabetes mellitus. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-4). IEEE.